

(51) 国際特許分類7
G06F 17/16

A1

(11) 国際公開番号

WO00/36527

(43) 国際公開日

2000年6月22日(22.06.00)

(21) 国際出願番号

PCT/JP99/07052

(22) 国際出願日

1999年12月15日(15.12.99)

(30) 優先権データ

特願平10/357796

1998年12月16日(16.12.98)

JP

(71) 出願人 (米国を除くすべての指定国について)

松下電器産業株式会社

(MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.)

[JP/JP]

〒571-8501 大阪府門真市大字門真1006番地 Osaka, (JP)

(72) 発明者 ; および

(75) 発明者 / 出願人 (米国についてのみ)

濱田真納(HAMADA, Mana)[JP/JP]

〒814-0022 福岡県福岡市早良区原1-40-1-201 Fukuoka, (JP)

九郎丸俊一(KUROMARU, Shunichi)[JP/JP]

〒814-0015 福岡県福岡市早良区室見3-1-1-401 Fukuoka, (JP)

米澤友紀(YONEZAWA, Tomonori)[JP/JP]

〒810-0014 福岡県福岡市中央区平尾5-19-8-105 Fukuoka, (JP)

中村 剛(NAKAMURA, Tsuyoshi)[JP/JP]

〒811-2413 福岡県糟屋郡篠栗町尾仲1100-1-405 Fukuoka, (JP)

(74) 代理人

弁理士 早瀬憲一(HAYASE, Kenichi)

〒564-0053 大阪府吹田市江の木町17番1号

江坂全日空ビル8階 早瀬特許事務所 Osaka, (JP)

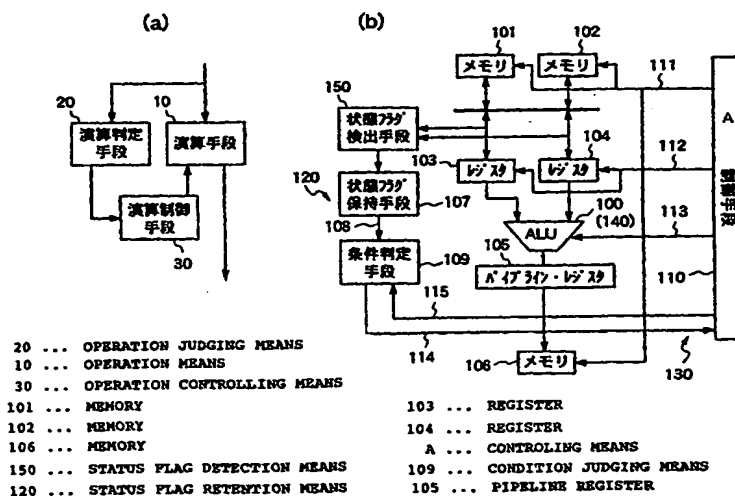
(81) 指定国 CN, IN, JP, US, 欧州特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE)

添付公開書類

国際調査報告書

(54) Title: CONDITIONAL VECTOR OPERATION METHOD AND DEVICE THEREOF

(54) 発明の名称 条件付きベクトル演算方法および条件付きベクトル演算装置



(57) Abstract

A status flag detection means (150) judges whether or not to operated on a first source data from memory (101) and retains the judged result as a status flag, and a condition judging means (109) judges whether or not the status flag satisfies a condition for permitting an operation. A controlling means (110) controls whether or not an ALU (100) should perform an operation based on this condition satisfied/unsatisfied information.

(57)要約

メモリ 101 からの第 1 のソースデータを状態フラグ検出手段 150 で演算を行うべきデータであるか否かを判定して、その判定結果を状態フラグとして保持し、条件判定手段 109 でその状態フラグが演算を行うべき条件が成立しているか否かを判定する。制御手段 110 はこの条件成立／不成立情報に基づき ALU100 が演算を行うべきか否かを制御する。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AE	アラブ首長国連邦	DM	ドミニカ	KZ	カザフスタン	RU	ロシア
AL	アルバニア	EES	エストニア	LC	セントルシア	RD	スーダン
AM	アルメニア	EES	スペイン	LI	セントビンセント・グレンディン	SE	スウェーデン
AT	オーストリア	FR	フランス	LK	スリランカ	SG	シンガポール
AU	オーストラリア	FR	フランス	LR	リベリア	SI	スロベニア
AZ	アゼルバイジャン	GB	英国	LS	レソト	SK	スロヴァキア
BA	ボスニア・ヘルツェゴビナ	GB	英国	LT	リトアニア	SL	シエラ・レオネ
BB	バルバドス	GD	グレナダ	LV	ルクセンブルグ	SN	セネガル
BE	ベルギー	GE	グルジア	LV	ラトヴィア	SZ	スワジランド
BF	ブルキナ・ファソ	GH	ガナ	MA	モロッコ	TD	チャド
BG	ブルガリア	GM	ギニア	MC	モナコ	TG	トゴ
BJ	ベナン	GN	ギニア・ビサウ	MD	モルドヴァ	TJ	タジキスタン
BR	ブラジル	GW	ギニア・ビサウ	MG	マダガスカル	TZ	タンザニア
BY	ベラルーシ	HR	クロアチア	MK	マケドニア旧ユーゴスラヴィア共和国	TM	トルクメニスタン
CA	カナダ	HU	ハンガリー	ML	マリ	TR	トルコ
CF	中央アフリカ	ID	インドネシア	MN	モンゴル	TT	トリニダード・トバゴ
CG	コンゴ	IE	アイルランド	MR	モーリタニア	UG	ウガンダ
CH	スイス	IL	イスラエル	MW	マラウイ	US	米国
CI	コートジボワール	IN	インド	MX	メキシコ	UZ	ウズベキスタン
CM	カメルーン	IS	アイスランド	NE	ニジェール	VN	ヴェトナム
CN	中国	IT	イタリア	NL	オランダ	YU	ユーゴスラビア
CR	コスタ・リカ	JP	日本	NO	ノルウェー	ZA	南アフリカ共和国
CU	キューバ	KE	ケニア	NZ	ニュージーランド	ZW	ジンバブエ
CY	キプロス	KG	キルギスタン	PL	ポーランド		
CZ	チェッコ	KP	北朝鮮	PT	ポルトガル		
DE	ドイツ	KR	韓国	RO	ルーマニア		
DK	デンマーク						

明 細 書

条件付きベクトル演算方法および条件付きベクトル演算装置

5 技術分野

本発明は、条件付きベクトル演算方法ならびに条件付きベクトル演算装置に関し、特に、マイクロプロセッサやディジタルシグナルプロセッサ（以下、DSP と称す）などと呼ばれるプロセッサやデータ処理装置に搭載される命令セットに用意された、ベクトル命令や
10 条件付き演算命令を演算する演算方法、ならびにこの演算方法を用いて演算を行う演算装置の改良を図ったものに関する。

背景技術

第 4 図は、マイクロプロセッサや DSP といった汎用的な命令セットを備えたデータ処理装置、あるいはこの種のデータ処理装置を搭載した、システム LSI と呼ばれている半導体集積回路装置を用い、演算対象のデータについて、例えば演算対象のデータがゼロであるなどの、或る条件が成立するか否かによって演算対象のデータに対して行うべき処理が異なってくるような処理を行う場合のフローチャートを示している。
20

この第 4 図では、まずステップ S401 において演算対象のデータ 402 を取得する。次にステップ S403 では、先のステップ S401 において取得したデータ 402 がゼロか否かを確認するためにゼロとデータ S402 との比較処理を行う。

25 次のステップ S404 では前のステップ S403 の結果を受けて、その比較結果が不一致であった（No）場合にはステップ S406 に移り処理 B を行うが、ステップ S404 による判定結果が一致であった（Yes）場合には、ステップ S405 で演算対象のデータ S402 に対して処理 A を行った後、ステップ S406 に移る。

ここで処理 B とは、演算対象のデータ、あるいは演算対象のデータに対して処理 A の演算を施したデータを用いた処理であるとする。

このように、第 4 図のフローチャートではステップ S404 の実行結果によって、以降の処理手順、即ちデータの処理の流れが異なってくる。

ところで、一般的なマイクロプロセッサや DSP などと呼ばれるプロセッサが汎用命令セットとして備えている命令の 1 つに条件分岐命令がある。条件分岐命令とは、実行すべき演算命令において演算結果の性質を示す情報をもったコンディション・コードと呼ばれるコードを前もって生成して保持しておき、このコンディション・コードが条件分岐命令の指定する条件と一致していれば条件成立として分岐を行い、一致していなければ条件不成立として分岐を行わないという命令である。従って、第 4 図のフローチャートにおけるステップ S404 のような、条件判断後の処理フローの分岐という処理を実行するには、条件分岐命令を用いるのが一般的である。

しかしながら、第 4 図に示したような処理を行う場合に必要となる条件分岐命令は、データ処理には直接的に何ら寄与しない動作であるにもかかわらず、これを `if ~ then ~ else` 文などで記述する必要があるが、この方法では 1 つの条件分岐に対応する `if` 文を処理するマイクロコードの実行に数サイクルの実行サイクルを要し、これをデータの個数だけ繰り返す必要があり、さらにプログラムをパイプライン制御しているようなシステム LSI においてはパイプラインの流れが中断されてしまい、これによる処理のオーバーヘッドが増加してしまう。

また、このような処理性能の低下を防ぐために、条件付き演算命令と呼ばれる命令を命令セットのなかに持つ機種がある。この条件付き演算命令は、様々な命令に分岐命令と同様な条件を付加し、指定された条件が成立したときのみその命令を実行し、不成立の場合は指定された動作を実行せずにそのまま次のステップに移るような

命令である。このような命令をもつことによって、第4図のステップ 404 のような条件分岐命令をプログラムから排除することができ、処理性能の低下を防ぐことが可能となる。

しかしながら、この条件付き演算を或るまとまった量のデータに
5 対し実行しようとする場合、予め、各データに関して条件を満足するか否かを算出しておき、これをフラグとして別途マスクレジスタに保持しておき、このフラグを参照しながら演算を実行するか否かを判定するようにしているが、この前処理としてのフラグ算出がオーバーヘッドとなってしまう、かつこのフラグを蓄えるマスクレジ
10 スタが別途必要になるという問題があった。

さらに、この種の条件付き演算命令の改良を図ったものとして、例えば特開平8-305563号公報に示されたデータ処理装置のように、演算命令に、無条件でこれを実行するか、条件が「真」の時にこれを実行するか、条件が「偽」の時にこれを実行するか、を
15 選択する少数のビットをデータに付加することにより、条件付き演算命令の命令の種類やオペランドの自由度の制限による性能低下を防止し、プログラムメモリの小容量化を可能にしたものもある。

しかしながら、演算対象のデータがゼロであれば処理 A として演算対象のデータに“1”を加算し、演算対象のデータがゼロでなければそのまま次の処理に移るような処理を、画像データのようなまとまったデータ数のデータに対して行いたい場合、従来の演算方法あるいは演算装置では、上述のように、第4図のフローを `if ~ then ~ else` 文などに相当するマイクロコードで記述しこれをデータの個数だけ繰り返すか、あるいは条件付き演算命令を用いて
20 フローのなかからステップ S404 を排除したものをデータの数だけ繰り返すか、のいずれかの方法で対処しなければならないが、これらの方法では命令をベクトル演算処理することができないため、データの個数分だけループによる繰り返し処理が必要になり、その分処理のオーバーヘッドが増加する。

また、一般的にいつて、上述のような条件付き演算を実行するよ
うな処理モジュールが頻繁に現われるプログラムは少ないかもしれ
ないが、例えば、画像 CODEC の信号処理のような処理モジュール
では、膨大なデータに対して一定の処理ルーチンを繰り返し実行す
ることが多く、処理ルーチンに含まれるステップ数が 1 ステップで
も増減することによる処理性能への影響は大きい。

本発明はこうした事情に鑑みてなされたもので、あるまとまった
データ量に対して上述したような条件付き演算を含む処理ルーチン
による処理を行う際に、条件付き演算をベクトル演算処理可能とし、
別途マスクレジスタを必要とせず、繰り返し処理に要する処理のオ
ーバーヘッドを削減でき、処理性能を向上することができる、条件
付きベクトル演算方法および条件付きベクトル演算装置を提供する
ことを目的とする。

15 発明の開示

上記目的を達成するために、本願の請求の範囲第 1 項の発明にか
かる条件付きベクトル演算方法は、演算処理を実行する演算処理工
程が演算処理の対象となるデータを取得する際に、これと並行して
当該演算を実行すべきか否かを算出し判定する演算判定工程と、前
記演算処理工程がベクトル演算を実行するように演算制御を行う際
に、前記演算判定工程の判定結果に応じて、前記演算処理の対象と
なるデータに対し演算を実行しその結果を出力するか、あるいは当
該データを演算を実行することなく出力するように演算制御を行う
演算制御工程とを含み、条件付き演算をベクトル演算処理可能にし
たことを特徴とする。

これらの処理工程によれば、条件付き演算を、演算を行うべきか
否かを判定し、その判定結果に応じて演算を実行させるか否かを制
御する処理と、この処理に応じて演算を行った結果もしくは演算を
行わない入力データそのものを出力する演算処理とに分割すること

ができ、これらを並行して動作でき、かつ、演算を行う場合も行わない場合もデータの流れが同一の経路を辿るので、条件付き演算を、パイプラインの処理の流れを中断することなく処理することが可能となり、これをベクトル演算にて高速に実行できる。

- 5 また、上記目的を達成するために、本願の請求の範囲第2項の発明にかかる条件付きベクトル演算方法は、ベクトル演算命令の発行によってデータの供給を開始するソースデータ供給処理工程と、これと並行して実行され、前記ソースデータ供給処理工程によって供給されるデータのうち所定のソースに供給されるデータについて逐
- 10 次その状態を保持し、状態フラグとして出力する状態フラグ保持処理工程とを有する第1のステージと、前記ソースデータ供給処理によって供給されるデータを用いて演算を行い演算結果を出力する演算処理工程と、これと並行して実行され、前記状態フラグと前記ベ
- 15 クトル演算命令によって発行される条件との条件判断を行い、条件成立または不成立の情報を制御処理工程に提供する条件判定処理工程と、を有する第2のステージと、前記演算処理工程による演算結果を順に格納する演算結果格納処理工程を有する第3のステージとを有するパイプライン処理を備え、前記パイプライン処理を実行することによりベクトル演算処理を実行するための制御を行うとともに、
- 20 前記条件判定処理の出力する情報が条件成立のときには前記演算処理を実行し、条件不成立のときには前記演算処理の演算結果として前記ソースデータ供給処理工程によって供給されるデータのうち前記所定のソースに供給されるデータをそのまま出力するように制御を行う制御処理工程を含む、ことを特徴とする。

- 25 これらの処理工程によれば、ソースデータ供給処理によりベクトル命令形式で次々と供給されるデータに対して、パイプラインの第1ステージでデータの性質（例えばゼロかゼロでないか）を状態フラグとして出力し、第2ステージでは条件判定処理工程によりその状態フラグの出力と命令の発行する条件とを用いて条件判定を行い、

判定結果を演算処理の制御に用いることが可能となる。つまり条件付き演算を条件判定と条件分岐演算のパイプライン・ステージを分けることによって、条件付き演算をベクトル演算形式で処理することができる。よって、この条件付きベクトル演算を第4図に示した

5 従来の処理フローに適用することで、ステップ S401 およびステップ S403 とステップ S404 さらにステップ S405 がパイプライン的に処理されることとなり、多数のデータを連続で処理した場合には繰り返しのルーチンが不要となり、処理性能が向上する。また、別途マスキレジスタを要することもなく、事前のフラグ判定も必要としないものが得られる。

10

また、上記目的を達成するために、本願の請求の範囲第3項の発明にかかる条件付きベクトル演算装置は、演算処理を実行する演算手段と、前記演算手段が演算処理の対象となるデータを取得する際にこれと並行して当該演算を実行すべきか否かを算出し判定する演算判定手段と、前記演算手段がベクトル演算を実行するよう演算制御を行う際に、前記演算判定手段の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算制御を行う演算制御手段とを備え、条件付き演算をベクトル演算処理可能にしたことを特徴とする。

15

20

この構成によれば、条件付き演算を、演算を行うべきか否かを判定し、その判定結果に応じて演算を実行させるか否かを制御する処理と、この処理に応じて演算を行った結果もしくは演算を行わない入力データそのものを出力する演算処理とに分割することができ、

25 これらを並行して動作でき、かつ、演算を行う場合も行わない場合もデータの流れが同一の経路を辿るので、条件付き演算を、パイプラインの流れを中断することなく処理することが可能となり、これをベクトル演算にて高速に実行できる。

また、上記目的を達成するために、本願の請求の範囲第4項の発

明にかかる条件付きベクトル演算装置は、ベクトル演算命令の発行によって第1ないし第Nのソースデータ（Nは2以上の整数）の供給を開始するソースデータ供給手段と、前記ソースデータ供給手段より供給される第1ないし第Nのソースデータを一時保存する第1
5 ないし第Nのレジスタと、前記第1ないし第Nのレジスタの出力を用いて演算を行う演算手段と、前記演算手段の出力する演算結果を一時保存するパイプライン・レジスタと、前記パイプライン・レジスタの出力を順次格納する演算結果格納手段と、前記第1ないし第Nの中の所定のソースデータの性質を表わす状態フラグ情報を逐次
10 保持する状態フラグ保持手段と、前記状態フラグ保持手段の出力と前記ベクトル演算命令によって指定される条件情報より、前記ベクトル演算命令の条件が成立か不成立かを示す情報を出力する条件判定手段と、前記ソースデータ供給手段がソースデータを供給し前記第1ないし第Nのレジスタに格納するとともに、これと並行して前
15 記状態フラグ保持手段が状態フラグ情報を保持し出力するまでを第1ステージ、前記演算手段が前記第1ないし第Nのレジスタの出力に対する演算処理結果を前記パイプライン・レジスタに出力するとともに、これと並行して前記条件判定手段が前記ベクトル演算命令の条件が成立か不成立かを示す情報を出力するまでを第2ステージ、
20 前記パイプライン・レジスタの出力を前記演算結果格納手段に格納するまでを第3ステージとしたパイプライン処理により、前記ベクトル演算命令を実行するための制御信号を生成するとともに、前記条件判定手段の出力する情報を受けて、条件不成立のときには前記演算手段の出力として前記第1ないし第Nの中の所定のソースデータの値をそのまま出力し、また条件成立のときには前記演算手段による演算結果を選択して出力するようなモード選択信号を生成する
25 制御手段とを備えたことを特徴とする。

この構成によれば、ソースデータ供給手段からベクトル命令形式で次々と供給されるデータに対して、パイプラインの第1ステージ

でデータの性質（例えばゼロかゼロでないか）を状態フラグ保持手段に格納し、第2ステージでは条件判定手段によりその状態フラグ保持手段の出力を命令の発行する条件と条件判定を行い、判定結果を演算手段の制御に用いることが可能となる。つまり条件判定と条件分岐演算のパイプライン・ステージを別のパイプライン・ステージに分けることによって、条件付き演算をベクトル演算形式で処理することができる。よって、この条件付きベクトル演算を第4図に示した従来の処理フローに適用することで、ステップ S401 およびステップ S403 とステップ S404 さらにはステップ S405 がパイプライン的に処理されることとなり、多数のデータを連続で処理した場合には繰り返しのルーチンが不要となり、処理性能が向上する。また、別途マスクレジスタを要することなく、事前のフラグ判定に要するオーバーヘッドも存在しないものが得られる。

15 図面の簡単な説明

第1図は、本発明の実施の形態1による条件付きベクトル演算装置の構成図であり、第1(a)図は、本願の請求の範囲第3項に対応する構成を示す図であり、第1(b)図は本発明の実施の形態1による条件付きベクトル演算装置のブロック構成を示す図である。

20 第2図は、本発明の実施の形態1による処理のフローチャートを示す図である。

第3図は、本発明の第1の実施の形態にかかる第2図のフローチャートに示した処理を本発明の第1の実施の形態の条件付きベクトル演算装置を用いて処理した場合のタイミングチャートを示す図である。

25 ある。

第4図は、従来の条件付き演算を含む処理のフローチャートを示す図である。

第5図は、第1(b)図における状態フラグ検出手段の構成例を示す図である。

第 6 図は、第 1 (b) 図に示す本発明の実施の形態 1 による条件付きベクトル演算装置のブロック構成の他の例を示す図である。

発明を実施するための最良の形態

5 以下、本発明の実施の形態について、第 1 図ないし第 3 図を用いて説明する。

(実施の形態 1)

この実施の形態 1 は、演算処理すべきデータの性質を判定して、このデータの性質に応じて演算を行うか否かを制御することにより、
10 条件付き演算をパイプライン処理、即ち、一連の処理を複数のステージに分け、ある一連の処理の第 1 のステージを実行し終わり第 2 のステージの実行中に、これと並行して次の一連の処理の第 1 のステージを実行する…、という処理を間断なく繰り返すベクトル演算処理、に組み込むことを可能にし、これにより、処理性能の向上を
15 図ったものである。

第 1 (a) 図は本発明の実施の形態 1 による条件付きベクトル演算装置の概略構成を示すものである。

第 1 (a) 図において、10 は演算処理を実行する演算手段、20 は演算手段 10 が演算処理の対象となるデータを取得する際にこれと並行して当該演算を実行すべきか否かを算出し判定する演算判定手段、30 は演算手段 10 がベクトル演算を実行するように演算制御を行う際に、演算判定手段 20 の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算
25 制御を行う演算制御手段である。

そして、これら演算手段 10、演算判定手段 20、演算制御手段 30 は、演算手段 10 が、演算処理を実行する演算処理工程に、演算判定手段 20 が、演算処理の対象となるデータを取得する際に、これと並行して当該演算を実行すべきか否かを算出し判定する演算

判定工程に、演算制御手段 30 が、演算処理工程がベクトル演算を実行するように演算制御を行う際に、演算判定工程の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算制御を行う演算制御工程に、それぞれ相当する処理を行うものである。

次に動作について説明する。演算手段 10 は図示しないメモリから演算の対象となるデータを取得し、これに演算命令に応じた演算処理を行おうとする。その際、演算判定手段 20 はこの演算の対象となるデータが演算処理すべきデータか否かを、演算手段 10 がデータを取得する際に、算出し判定する。

そしてその判定結果に基づいて演算制御手段 30 は演算処理を行うか否かを決定し、演算を行うべきデータと判定した場合は当該データを命令に応じて処理し、また、否と判定した場合は、演算を実行せずにそのままデータを出力するように、演算手段 10 を制御する。

演算制御手段 30 は、演算手段 10 がベクトル演算命令に対し、パイプライン処理を実行するように、即ち、データの供給、演算、演算結果の格納、の各ステージを時間的にずらせながら並行して実行することにより、ベクトル命令を処理するように制御を行っているが、上述のように、条件付き演算命令の条件判定を本来のパイプライン処理と並行して行い、かつ演算を行う場合も行わない場合もデータの流れが同一の経路を辿るようにしたので、条件付き演算をベクトル演算として処理することが可能となり、ベクトル演算化していない条件付き演算により大量のデータを繰り返し演算処理する場合のオーバーヘッドを軽減することが可能となる。

第 1 (b) 図は、この実施の形態 1 による条件付きベクトル演算装置の構成をより具体的に示したものである。

第 1 (b) 図において、100 は本発明を適用した条件付きベクト

ル演算装置において、その演算処理を実行する ALU であり、第 1 (a) 図の演算手段 10 に対応する。また、101 および 102 はソースデータ供給手段としてのメモリ、103 および 104 はそれぞれメモリ 101 と 102 より供給されるソースデータを一時保存するレジスタ、105 は ALU100 が出力する演算結果を一時的に保存するためのパイプライン・レジスタ、106 はパイプライン・レジスタ 105 の出力を順次格納するための演算結果格納手段としてのメモリ、150 はメモリ 101 から読み出されるソースデータの性質を表わす情報を逐次検出する状態フラグ検出手段、107 は状態フラグ検出手段 150 が検出する状態フラグを逐次保持する状態フラグ保持手段、108 は状態フラグ保持手段 107 が出力する状態フラグ、120 はこれら状態フラグ検出手段 150 および状態フラグ保持手段 107 からなる演算判定手段であり、第 1 (a) 図の演算判定手段 20 に対応し、演算手段が演算処理の対象となるデータを取得する際にこれと並行して当該演算を実行すべきか否かを算出し判定する。

また、109 は状態フラグ 108 と後述する条件情報 115 を入力とし、条件判定を行って条件の成立／不成立を示す情報 114 を出力する条件判定手段、110 はベクトル演算命令の発行を受けて本発明の演算装置全体をベクトル命令形式でパイプライン制御するように制御信号を生成する制御手段、130 はこれら条件判定手段 109 および制御手段 110 からなる演算制御手段であり、第 1 (a) 図の演算制御手段 30 に対応し、演算手段がベクトル演算を実行するよう演算制御を行う際に、前記演算判定手段の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算制御を行う。ALU100 はこの演算制御手段 130 の制御に応じて演算処理すべきデータに対し演算を実行して出力するかあるいは演算を実行することなく出力する演算手段 140 として動作する。111 は制御手段 110 が出力するメモリ 101、102、106 のメモリ制御信号、112 は制御手

段 110 が出力するレジスタ 103,104 のライト・イネーブル信号、113 は制御手段 110 が出力する ALU100 のモード選択信号、114 は条件判定手段 109 が出力する条件成立／不成立を示す情報、115 は制御手段 110 が出力する命令による条件情報である。

- 5 ここで、本実施の形態 1 による条件付きベクトル演算装置の構成を用いた場合の動作を、第 2 図にフローチャートを示した処理を行う場合を例にとりて説明する。

- 第 2 図では、まずステップ S200 において j を“0”に初期化する。次にステップ S201 において、この j を“0”に初期化した状態から
10 演算対象のデータ X(j)をメモリ 101 から 取得する。このステップ S201 はソースデータ供給処理に対応する。次にステップ S202 では、前のステップ S201 において取得したデータ X(j)がゼロか否かを確かめるために、ゼロとデータ X(j)との比較処理を行い、その比較結果を保持する。このステップ S202 は状態フラグ保持処理工程に相当する。
15 実際にこれらステップ S201 およびステップ S202 は並行して処理がなされ、これらステップ S201 およびステップ S202 により第 1 (b) 図のメモリ 101 から演算対象となるデータを読み出す第 1 ステージ 310 が構成されている。この第 1 ステージ 310 は演算判定工程に相当するものである。

- 20 次に、ステップ S203 では条件付き演算命令を用い、前のステップ S202 において決定したコンディション・フラグよりその比較結果が不一致であった (No) 場合には、ステップ S204 により X(j)をそのまま Y(j)として出力し、比較結果が一致した (Yes) 場合には、ステップ S205 により X(j)に“1”を加算したものを Y(j)として出力し、
25 ステップ S210 に移る。実際にこれらステップ S203 およびステップ S204 .S205 は並行して処理がなされ、これらステップ S203 およびステップ S204 .S205 により演算処理を行う第 2 ステージ 311 が構成されている。この第 2 ステージ 311 は演算処理工程と演算制御工程に相当するものであり、ステップ S203 は条件判定処理工程に、ス

テップ S204 およびステップ S205 は演算処理工程にそれぞれ相当する。

次に、ステップ S210 ではこのようにして得られた演算結果をメモリ 106 に格納し、ステップ S206 に移る。このステップ S210 により
5 演算結果をメモリ 106 に書き込む第 3 ステージ 312 が構成されており、これは演算結果格納処理工程に相当する。

ステップ S206 は演算対象のデータである 8 個のデータの処理が終わるまで、ステップ S201 ないしステップ S205 の処理を繰り返すためのものであり、ステップ S206 において j が “8” 未満の (No)
10 場合には、ステップ S207 において j に “1” を加算し、ステップ S201 以降の処理を繰り返す。これら S206 およびステップ S207 により、第 1 ステージ 310 ないし第 3 ステージ 312 を有するパイプライン処理を備え、前記パイプライン処理を実行することによりベクトル演算処理を実行するための制御を行うとともに、前記条件判定処理の
15 出力する情報が条件成立のときには前記演算処理を実行し、条件不成立のときには前記演算処理の演算結果として前記ソースデータ供給処理によって供給されるデータのうち前記所定のソースに供給されるデータをそのまま出力するように制御を行う制御処理工程が実現されている。また、 j が “8” 以上の値となった (Yes) 場合には、
20 ステップ S208 において $Y(0)$ ないし $Y(7)$ の値の累算を行う。

このフローチャートは、 $X(j)$ で表わされた演算対象データについて、その値が “0” であるか否かによる条件分岐を行い、“0” 以外であった場合 (No) には演算結果 $Y(j)$ としては $X(j)$ そのものを、“0” であった場合 (Yes) には $Y(j)$ としては $X(j)+1$ を、それぞれ採用し、以上の処理を j が “8” になるまでの間繰り返すことによって、8 個の演算対象データに対し同様の条件付き演算を施す場合を示している。
25

以上の処理に、本実施の形態 1 の構成を用いて、条件付きベクトル演算を実行したときのタイミング・チャートを第 3 図に示した。

第 3 図において、301 は第 1 のソースデータとしての、メモリ 101

からのリードデータ、302 は第 2 のソースデータとしての、レジスタ 104 に設定した固定値 “1”、303 はレジスタ 103 の出力、304 は状態フラグ保持手段 107 の出力としての状態フラグ 108、305 は命令による条件情報 115 を示す値、306 は条件判定手段 109 の出力としての条件成立／不成立を示す情報 114、307 は ALU100 のモード選択信号 113、308 は ALU100 の出力、309 はパイプライン・レジスタ 105 の出力である。

また、310、311、312 は前述した通りの第 1、第 2、第 3 ステージである。即ち、310 はベクトル演算命令の発行によってソースデータの供給を開始するソースデータ供給処理としての第 1 ステージ、311 は前記ソースデータ供給処理によって供給されるデータを用いて演算を行い演算結果を出力する演算処理としての第 2 ステージ、312 は前記演算結果を順に格納する演算結果格納処理としての第 3 ステージであり、第 1 図の制御手段 110 はこれら第 1 ステージ 310、第 2 ステージ 311、第 3 ステージ 312 をパイプライン処理として実行するように、制御信号を生成している。

ここで第 3 図における状態フラグ 304 は、第 1 のソースデータ 301 のデータの性質の情報であり、データがゼロの場合には “1” の値を、データがゼロ以外の場合には “0” の値を、それぞれ出力するものとする。また、命令による条件情報 305 としては、“ゼロである” という条件のコンディション・コードは “1” であるものとする。また ALU100 のモード選択信号が、演算処理を行わない “スルー” のときには ALU100 の出力 306 には第 1 のソースデータ 301 を、“ADD” 演算を行うときには第 1 のソースデータ 301 と第 2 のソースデータ 302 の値を加算した結果を出力するものとする。

即ち、第 3 図のサイクル 1 において、制御手段 110 は第 1 図のレジスタ 104 に値 “1” を設定するとともに、命令による条件情報 305 として値 “1” を設定する。次に、サイクル 2 において、制御手段 110 はメモリ 101 よりデータ X(0)として値 “1” を読み出し、これをレジ

スタ 103 および状態フラグ検出手段 150 に転送する。状態フラグ保持手段 107 はこの状態フラグ検出手段 150 による検出結果“0”を保持する。次に、サイクル 3 において、条件判定手段 109 はこの状態フラグ保持手段 107 に保持されている値 108 が“0”であり、かつ制御手段 110 が出力する、命令による条件情報 115 の値が“1”であるため、これらと比較し、条件が不成立である旨の条件成立／不成立情報 114 を制御手段 110 に出力する。これにより、制御手段 110 は ALU100 に対しスルーを選択する旨の ALU のモード選択信号 113 を出力する。これにより、ALU100 は、第 1 のソースデータとしてのレジスタ 103 の出力に対し何ら演算を行うことなくその値“1”をそのまま出力する。次に、このサイクル 4 において、ALU100 の出力“1”は、制御手段 110 によりパイプライン・レジスタ 105 に転送され、このパイプライン・レジスタ 105 の出力は、メモリ 106 に書き込まれる。

これらサイクル 2 ないし 4 において、データ X(0)がスルー処理されるのと並行して、値“2”を有するデータ X(1)がこれらより 1 サイクルずつずれたサイクル 3 ないし 5 において、同様に処理される。この場合も条件が不成立であるので、ALU100 は、その値“2”をそのまま出力する。

次に、サイクル 4 において、制御手段 110 はメモリ 101 よりデータ X(2)として値“0”を読み出し、これをレジスタ 103 および状態フラグ検出手段 150 に転送する。状態フラグ保持手段 107 はこの状態フラグ検出手段 150 による検出結果“1”を保持する。次にサイクル 5 において、条件判定手段 109 はこの状態フラグ保持手段 107 に保持されている値 108 が“1”であり、かつ制御手段 110 が出力する、命令による条件情報 115 の値が“1”であるため、これらと比較し、条件が成立している旨の条件成立／不成立情報 114 を制御手段 110 に出力する。これにより、制御手段 110 は ALU100 に対し加算演算を選択する旨の ALU のモード選択信号 113 を出力する。これにより、

ALU100 は、第 1 のソースデータとしてのレジスタ 103 の出力 “0”
に対し “1” を加える演算を行ってその加算値 “1” を出力する。次
に、サイクル 6 において、この ALU100 の出力 “1” は、制御手段
110 によりパイプライン・レジスタ 105 に転送され、このパイプ
5 ライン・レジスタ 105 の出力は、メモリ 106 に書き込まれる。

以下、このようなパイプライン動作が値 “1”, “2”, “3”, “0”, “5”
を有するデータ X(3), X(4), X(5), X(6), X(7)に対しても同様に行
われ、パイプラインレジスタ 105 の出力として、値 “1”, “2”, “3”,
“1”, “5” がそれぞれ得られる。

10 これらの動作を処理の流れに着目して述べると、第 1 ステージ 310
では、メモリ 101 から第 1 のソースデータとしてのリードデータ 301
を得るとともに、第 2 のソースデータとして、レジスタ 104 に固定
値 “1”を設定し、かつベクトル演算命令による条件情報 115 の値 “1”
を設定する。そして、メモリ 101 から第 1 のソースデータとしての
15 リードデータ 301 をレジスタ 103 に転送するとともに、その値を状
態フラグ検出手段 150 で “0”か否かを判定し、“0”以外であれば “0”
を、“0”であれば “1”を、それぞれ状態フラグとして状態フラグ保
持手段 107 で保持する。これらの処理において、メモリ 101 からの
リードデータ 301 の転送はソースデータ供給処理工程に相当し、状
20 態フラグ検出手段 150 および状態フラグ保持手段 107 の処理は演算
判定工程あるいは状態フラグ保持処理工程に相当する。

次に、第 2 ステージ 311 では、条件判定手段 109 で状態フラグ保
持手段 107 で保持している状態フラグの値と制御手段 110 から出力
される、ベクトル演算命令による条件情報 115 の値 “1”とを比較し、
25 ALU100 による演算を行うべきか否かを示す条件成立／不成立情報
114 を制御手段 110 に出力する。制御手段 110 はこの条件成立／不
成立情報 114 に応じて ALU100 に対し ALU のモード選択信号 113
を出力し、条件が成立した場合に ALU100 がレジスタ 103 の出力に
対しレジスタ 104 の出力である固定値 “1”を加える演算を行い、条

件が不成立の場合にレジスタ 103 の出力に何も演算を行わず、これをスルーで出力するように制御を行い、この ALU100 の出力をパイプラインレジスタ 105 に出力する。これらの処理において、条件判定手段 109 による処理は条件判定処理工程に相当し、ALU100 の処理は演算処理工程に相当し、条件判定手段 109 および制御手段 110 の処理は演算制御工程に相当する。

そして、第 3 ステージ 312 では、ALU100 の出力を一時格納したパイプラインレジスタ 105 の出力を、累算値を得るためのメモリ 106 に書き込む。このメモリ 106 への書き込みは演算結果格納処理工程に相当する。

この第 3 図に示したタイミング・チャートにより、データに対し演算を行う場合も行わない場合もデータの流れる経路を辿るので、条件付き演算を、パイプライン処理の流れを中断することなく実行できることが明らかとなり、従って、第 2 図のフローチャートで述べた処理のうちステップ S201 ないしステップ S207 を 8 回繰り返す分のステップがベクトル演算で実現できることが明らかとなった。このため、条件付き演算をベクトル演算に含めることができ、これに通常のベクトル演算と同様、1 命令を割り当てることができ、従って、これらサイクル 1 ないしサイクル 11 に相当するパイプライン処理を 1 命令で実現でき、しかもその処理を、(8 サイクル) + (パイプライン遅延段数) という処理量で実現できる。

このように、本実施の形態 1 によれば、処理ルーチン中に演算対象のデータの性質によって処理すべき演算内容が変わってくるような条件付き演算が含まれ、その演算対象が多数のデータにわたる場合に、条件付き演算を、条件の判定と条件分岐および演算のパイプライン・ステージを分けることにより、即ち、ALU により条件付き演算を行う際に、演算を実行するサイクルとは別に、演算のソースとなる第 1 のソースデータに基づき、演算を行う条件が成立しているか否かを判定し、この判定結果に基づいて演算を実行するか否か

を決定することにより、条件付き演算を、処理の流れを中断することなくベクトル演算形式で実行することが可能となり、これにより、演算対象データを連続的に処理することが可能となり、処理ステップ数の削減によるプロセッサの性能向上を図ることができる。また、
5 別途マスキレジスタ等のハードウェアも不要であり、消費電力に制約のある機器にプロセッサを組み込む場合にはその分有利となる。さらに、事前にフラグを算出することによるオーバーヘッドも生じない。

なお、以上の説明では、ソースデータ供給手段としてメモリ
10 101,102を用いたが、ソースデータ供給手段の一方をレジスタで構成した固定値としても同様の条件付きベクトル演算装置が実現可能である。

また、演算手段としてのALU100を累算回路で構成し、演算結果格納手段としてのメモリ106を削除し、パイプライン・レジスタ105
15 をアキュムレート・レジスタとして構成することにより、条件付きベクトル演算の累算装置を実現することも可能である。

また、ソースデータの性質を表わす情報を逐次保持する状態フラグ保持手段107に保持する状態フラグとしては、メモリ101または102のいずれか一方がゼロかゼロ以外かを表わすゼロ・フラグ
20 の他、正、負を表わすサイン・フラグ、メモリ101,102からの読み出しデータが一致しているか否かを表す一致フラグ、メモリ101,102からの読み出しデータの大小関係を表すフラグ、即ち、メモリ101からの読み出しデータとメモリ102からの読み出しデータとの大小関係が“ \geq ”, “ \leq ”, “ $>$ ”, “ $<$ ”のいずれの状態
25 になっているかを示すフラグなどを保持し、分岐の条件とすることが可能である。

また、第5図に示すように、状態フラグ検出手段150を、設定レジスタ150aおよび比較手段150bにより構成し、設定レジスタ150aに所望の設定値Aを入力可能とし、かつ比較手段15

0 bによりこの設定レジスタ 1 5 0 a の設定値 A と、メモリ 1 0 1, 1 0 2 のいずれか一方から読み出したデータとを比較するように構成することにより、これらの大小関係を表すフラグ、即ち、設定値 A と、メモリ 1 0 1, 1 0 2 のいずれか一方から読み出したデータとの大小関係が “ \geq ”, “ \leq ”, “ $>$ ”, “ $<$ ”, “ $=$ ” のいずれの状態になっているかを示すフラグなどを保持し、条件分岐の条件とすることも可能である。

さらに第 1 ステージの処理時間に収まる範囲で処理可能であれば、比較のロジックをより複雑な演算にすることにより、より複雑な分岐の条件に対応することも可能である。

また、第 6 図に示すメモリ 1 0 1 a ないし 1 0 1 n、あるいは 1 0 2 a ないし 1 0 2 n のように、メモリ 1 0 1, 1 0 2 の少なくとも一方を、複数のメモリから構成し、これらメモリ 1 0 1, 1 0 2 をソースデータ供給手段としてその出力を演算する場合、メモリ 1 0 1, 1 0 2 のいずれか一方のメモリの出力と他方のメモリの出力とを比較して、上述のような各種のフラグを保持し、条件分岐の条件とすることも可能である。この場合、レジスタ 1 0 3, 1 0 4 および ALU100 の入力にはメモリの増加に対応させて増設する必要がある。また、メモリ 1 0 1, 1 0 2 の出力はいずれか一方の 1 つの出力を第 1 のソースデータとし、他方の複数の出力を第 2 ないし第 N (N は 2 以上の整数) のソースデータとしてこれらと比較する必要がある。

さらに、膨大なデータに対して一定の処理ルーチンを繰り返し実行する処理モジュールであれば、画像データ以外のデータに対しても、その処理に使用することが可能である。

産業上の利用可能性

請求の範囲第 1 項の条件付きベクトル演算方法によれば、演算処理を実行する演算処理工程が演算処理の対象となるデータを取得す

る際に、これと並行して当該演算を実行すべきか否かを算出するとともに判定し、前記演算処理工程がベクトル演算を実行するように演算制御を行う際に、前記判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは
5 当該データを演算を実行することなく出力するように演算制御を行うようにしたので、条件付き演算がベクトル演算処理可能となり、ベクトル演算にて高速に実行できる。

請求の範囲第2項の条件付きベクトル演算方法によれば、ベクトル演算命令の発行によってデータの供給を開始するソースデータ供給
10 処理工程と、これと並行に実行され、前記ソースデータ供給処理工程によって供給されるデータのうち所定のソースに供給されるデータについて逐次その状態を保持し、状態フラグとして出力する状態フラグ保持処理工程とを有する第1のステージと、前記ソースデータ供給処理によって供給されるデータを用いて演算を行い演算結果
15 を出力する演算処理工程と、これと並行して実行され、前記状態フラグと前記ベクトル演算命令によって発行される条件との条件判断を行い、条件成立または不成立の情報を制御処理工程に提供する条件判定処理工程と、を有する第2のステージと、前記演算処理工程による演算結果を順に格納する演算結果格納処理工程を有する第
20 3のステージを有するパイプライン処理を備え、前記パイプライン処理を実行することによりベクトル演算処理により実行するための制御に加えて、前記条件判定処理の出力する情報が条件成立のときには前記演算処理を実行し、条件不成立のときには前記演算処理の演算結果として前記ソースデータ供給処理工程によって供給される
25 データのうち前記所定のソースに供給されるデータをそのまま出力するように制御を行う制御処理工程を含むようにしたので、条件付き演算を条件判定と条件分岐演算のパイプライン・ステージを分けることができ、条件付き演算をベクトル演算形式で処理することができる。このため、多数のデータを連続で処理した場合には繰り返し

しのルーチンが不要となり、処理性能が向上する。また、別途マスクレジスタ等のハードウェアも不要であり、事前にフラグを算出することによるオーバーヘッドも生じない。

請求の範囲第3項の条件付きベクトル演算装置によれば、演算手段が演算処理の対象となるデータを取得する際にこれと並行して当該演算を実行すべきか否かを演算判定手段により算出するとともに判定し、前記演算手段がベクトル演算を実行するように演算制御を行う際に、前記演算判定手段の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算制御を行うようにしたので、条件付き演算がベクトル演算処理可能となり、ベクトル演算にて高速に実行できる。

請求の範囲第4項の条件付きベクトル演算装置によれば、ベクトル演算命令の発行によって第1ないし第Nのソースデータ（Nは2以上の整数）の供給を開始するソースデータ供給手段と、前記ソースデータ供給手段より供給される第1ないし第Nのソースデータを一時保存する第1ないし第Nのレジスタと、前記第1ないし第Nのレジスタの出力を用いて演算を行う演算手段と、前記演算手段の出力する演算結果を一時保存するパイプライン・レジスタと、前記パイプライン・レジスタの出力を順次格納する演算結果格納手段と、前記第1ないし第Nの中の所定のソースデータの性質を表わす状態フラグ情報を逐次保持する状態フラグ保持手段と、前記状態フラグ保持手段の出力と前記ベクトル演算命令によって指定される条件情報より、前記ベクトル演算命令の条件が成立か不成立かを示す情報を出力する条件判定手段と、前記ソースデータ供給手段がソースデータを供給し前記第1ないし第Nのレジスタに格納するとともに、これと並行して前記状態フラグ保持手段が状態フラグ情報を保持し出力するまでを第1ステージ、前記演算手段が前記第1ないし第Nのレジスタの出力に対する演算処理結果を前記パイプライン・レジ

スタに出力するとともに、これと並行して前記条件判定手段が前記ベクトル演算命令の条件が成立か不成立かを示す情報を出力するまでを第2ステージ、前記パイプライン・レジスタの出力を前記演算結果格納手段に格納するまでを第3ステージとしたパイプライン処理により、前記ベクトル演算命令を実行するための制御信号を生成するとともに、前記条件判定手段の出力する情報を受けて、条件不成立のときには前記演算手段の出力として前記第1ないし第Nの中の所定のソースデータの値をそのまま出力し、また条件成立のときには前記演算手段による演算結果を選択して出力するようなモード選択信号を生成する制御手段とを備え条件付き演算を条件判定と条件分岐演算のパイプライン・ステージを分けることができ、条件付き演算をベクトル演算形式で処理することができる。このため、多数のデータを連続で処理した場合には繰り返しのルーチンが不要となり、処理性能が向上する。また、別途マスクレジスタ等のハードウェアも不要であり、事前にフラグを算出することによるオーバーヘッドも生じない。

20

25

請 求 の 範 囲

1. 演算処理を実行する演算処理工程が演算処理の対象となるデータを取得する際に、これと並行して当該演算を実行すべきか否かを算出し判定する演算判定工程と、

前記演算処理工程がベクトル演算を実行するように演算制御を行う際に、前記演算判定工程の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算制御を行う演算制御工程とを含み、

条件付き演算をベクトル演算処理可能にしたことを特徴とする条件付きベクトル演算方法。

2. ベクトル演算命令の発行によってデータの供給を開始するソースデータ供給処理工程と、これと並行して実行され、前記ソースデータ供給処理によって供給されるデータのうち所定のソースに供給されるデータについて逐次その状態を保持し、状態フラグとして出力する状態フラグ保持処理工程とを有する第1のステージと、

前記ソースデータ供給処理工程によって供給されるデータを用いて演算を行い演算結果を出力する演算処理工程と、これと並行して実行され、前記状態フラグと前記ベクトル演算命令によって発行される条件との条件判断を行い、条件成立または不成立の情報を制御処理工程に提供する条件判定処理工程と、を有する第2のステージと、

前記演算処理工程による演算結果を順に格納する演算結果格納処理工程を有する第3のステージとを有するパイプライン処理を備え、前記パイプライン処理を実行することによりベクトル演算処理を実行するための制御を行うとともに、

前記条件判定処理の出力する情報が条件成立のときには前記演算処理を実行し、条件不成立のときには前記演算処理の演算結果とし

て前記ソースデータ供給処理工程によって供給されるデータのうち前記所定のソースに供給されるデータをそのまま出力するように制御を行う制御処理工程を含む、

ことを特徴とする条件付きベクトル演算方法。

5 3. 演算処理を実行する演算手段と、

前記演算手段が演算処理の対象となるデータを取得する際にこれと並行して当該演算を実行すべきか否かを算出し判定する演算判定手段と、

10 前記演算手段がベクトル演算を実行するよう演算制御を行う際に、前記演算判定手段の判定結果に応じて、前記演算処理の対象となるデータに対し演算を実行しその結果を出力するか、あるいは当該データを演算を実行することなく出力するように演算制御を行う演算制御手段とを備え、

15 条件付き演算をベクトル演算処理可能にしたことを特徴とする条件付きベクトル演算装置。

4. ベクトル演算命令の発行によって第1ないし第Nのソースデータ(Nは2以上の整数)の供給を開始するソースデータ供給手段と、

前記ソースデータ供給手段より供給される第1ないし第Nのソースデータを一時保存する第1ないし第Nのレジスタと、

20 前記第1ないし第Nのレジスタの出力を用いて演算を行う演算手段と、

前記演算手段の出力する演算結果を一時保存するパイプライン・レジスタと、

25 前記パイプライン・レジスタの出力を順次格納する演算結果格納手段と、

前記第1ないし第Nの中の所定のソースデータの性質を表わす状態フラグ情報を逐次保持する状態フラグ保持手段と、

前記状態フラグ保持手段の出力と前記ベクトル演算命令によって指定される条件情報より、前記ベクトル演算命令の条件が成立か不

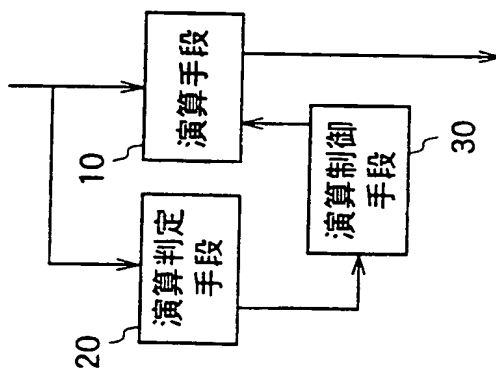
成立かを示す情報を入力する条件判定手段と、

- 前記ソースデータ供給手段がソースデータを供給し前記第 1 ないし第 N のレジスタに格納するとともに、これと並行して前記状態フラグ保持手段が状態フラグ情報を保持し出力するまでを第 1 ステージ、前記演算手段が前記第 1 ないし第 N のレジスタの出力に対する演算処理結果を前記パイプライン・レジスタに出力するとともに、これと並行して前記条件判定手段が前記ベクトル演算命令の条件が成立か不成立かを示す情報を入力するまでを第 2 ステージ、前記パイプライン・レジスタの出力を前記演算結果格納手段に格納するまでを第 3 ステージとしたパイプライン処理により、前記ベクトル演算命令を実行するための制御信号を生成するとともに、

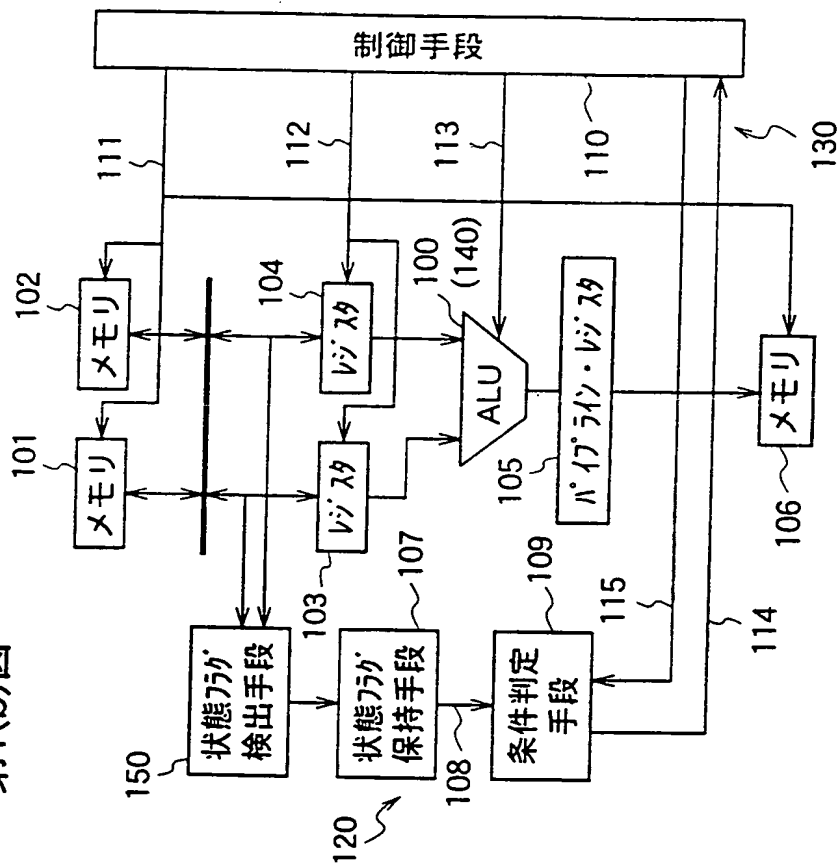
- 前記条件判定手段の出力する情報を受けて、条件不成立のときには前記演算手段の出力として前記第 1 ないし第 N の中の所定のソースデータの値をそのまま出力し、また条件成立のときには前記演算手段による演算結果を選択して出力するようなモード選択信号を生成する制御手段とを備えたことを特徴とする条件付きベクトル演算装置。

This Page Blank (uspto)

第1(a)図

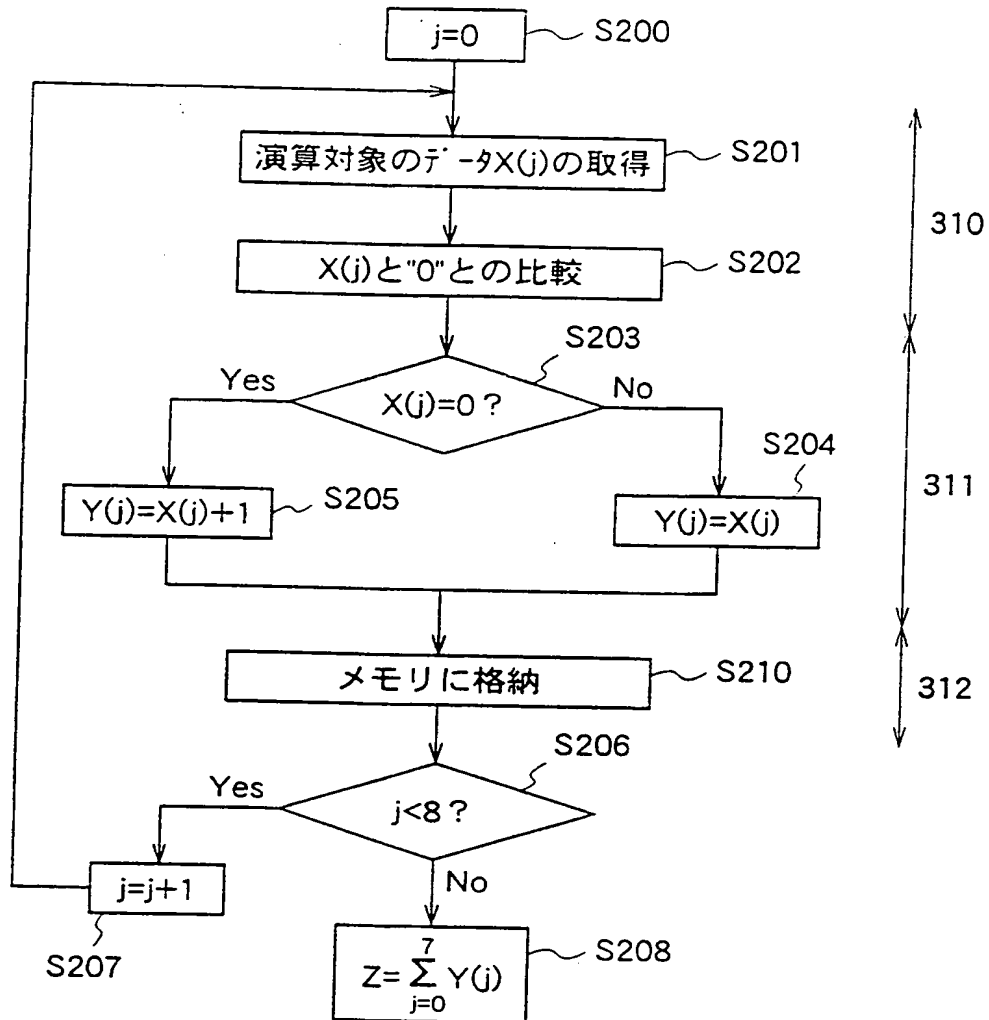


第1(b)図

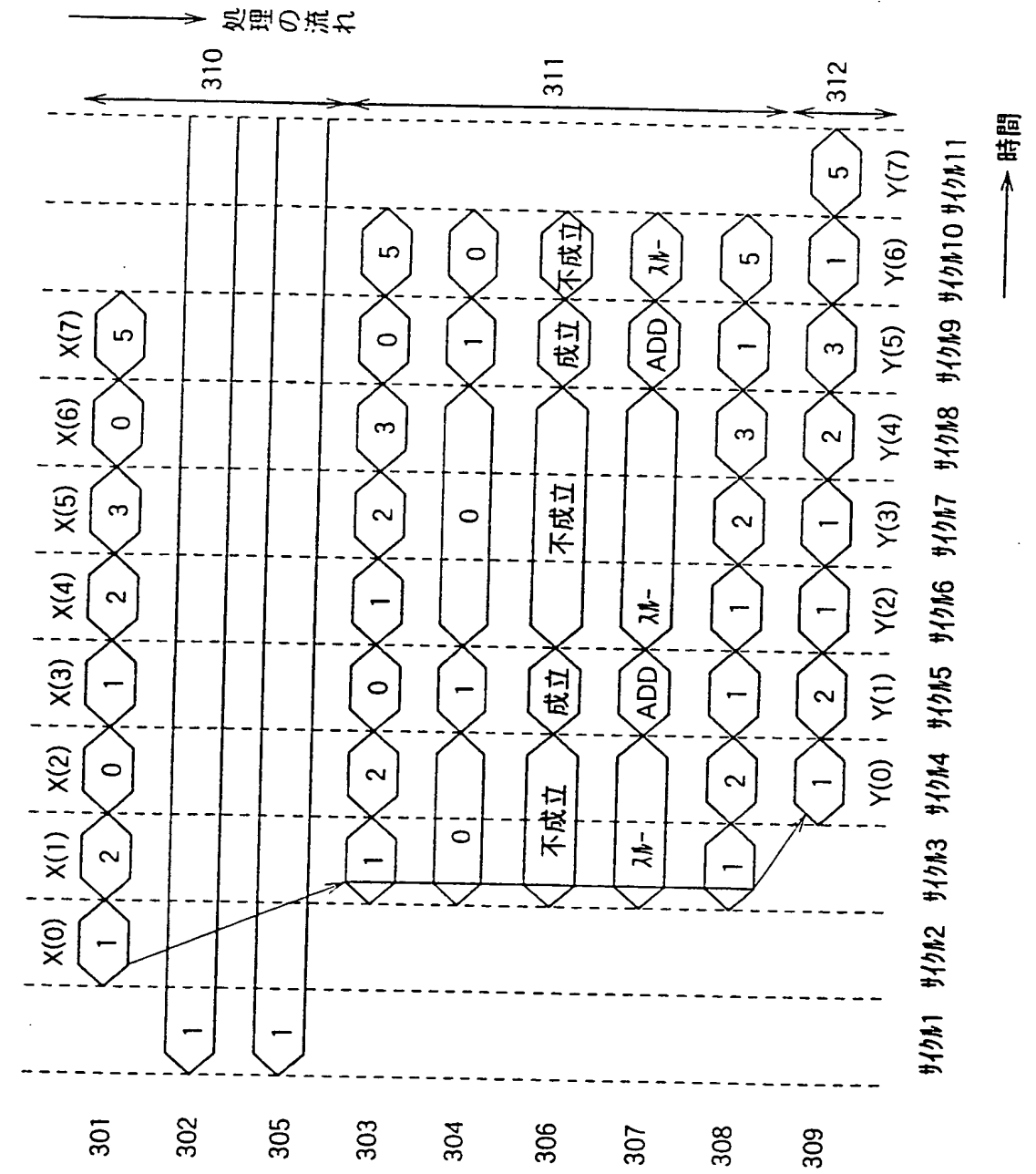


This Page Blank (uspto)

第2図

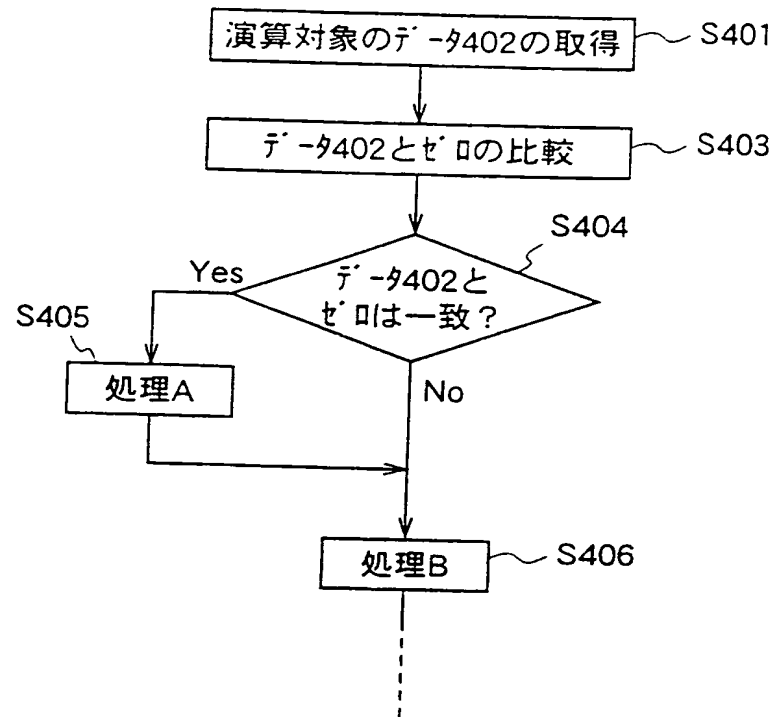


This Page Blank (uspto)



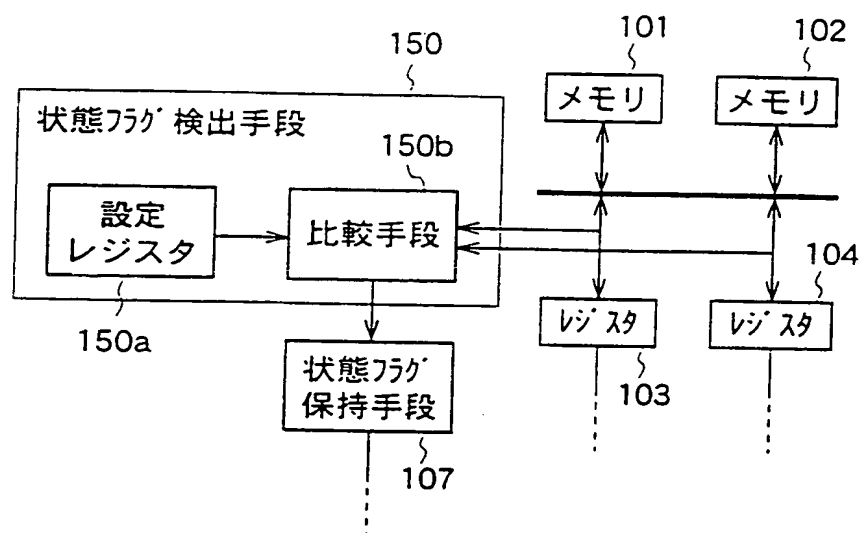
This page blank (uspto)

第4図

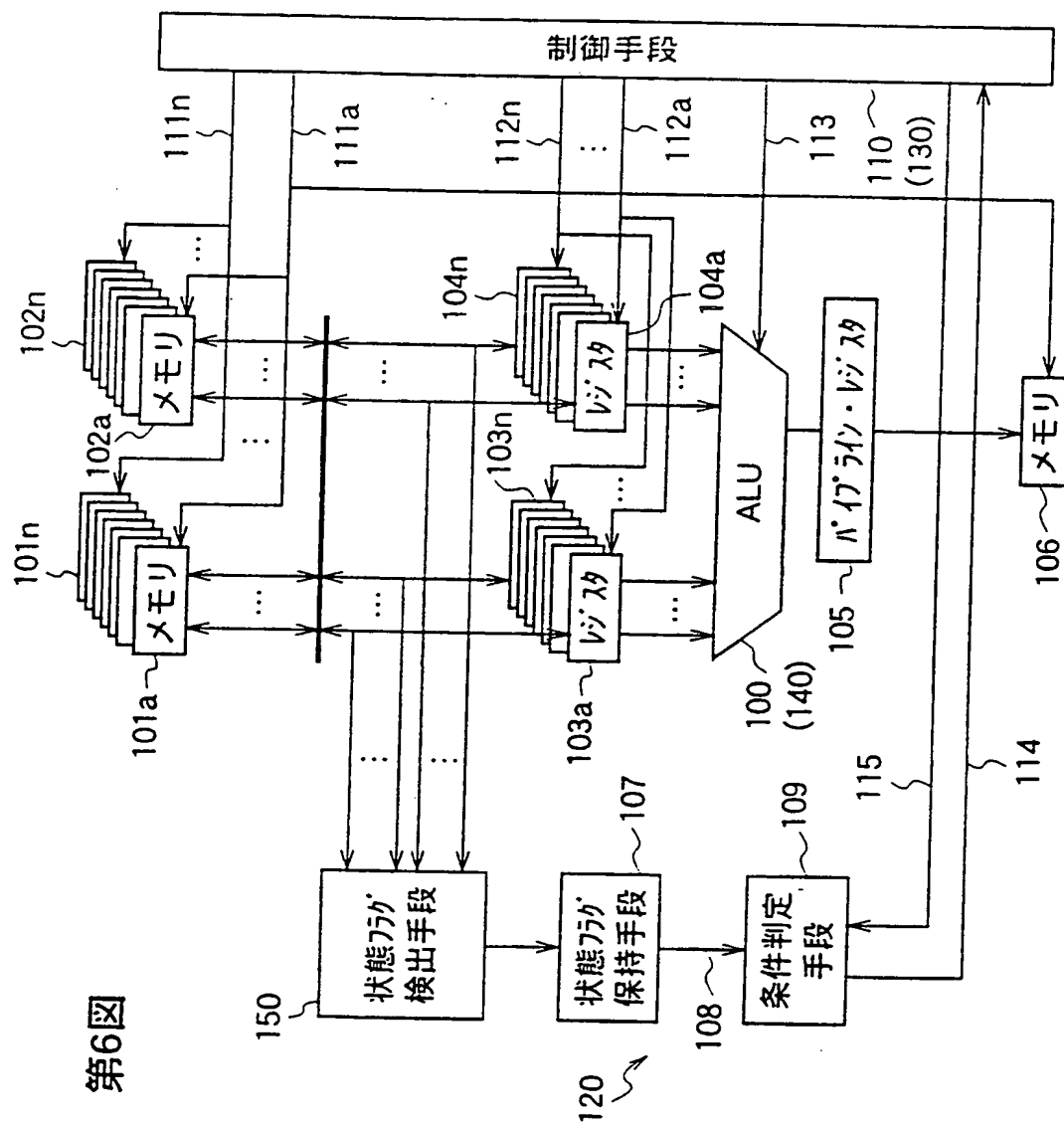


This Page Blank (uspto)

第5図



This Page Blank (uspto)



This Page Blank (uspto)